



Windows PowerShell Get-Help on Cmdlet 'New-AzTag'

PS:\>Get-HELP New-AzTag -Full

NAME

New-AzTag

SYNOPSIS

Creates a predefined Azure tag or adds values to an existing tag | Creates or updates the entire set of tags on a resource or subscription.

SYNTAX

New-AzTag [-Name] <System.String> [[-Value] <System.String>] [-DefaultProfile

<Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [-Confirm] [-WhatIf]
[<CommonParameters>]

New-AzTag [-ResourceId] <System.String> [-Tag] <System.Collections.Hashtable> [-DefaultProfile

<Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [-Confirm] [-WhatIf]
[<CommonParameters>]

DESCRIPTION

CreatePredefinedTagSet : The New-AzTag cmdlet creates a predefined Azure tag with an optional predefined value. You

can also use it to add additional values to

existing predefined tags. To create a predefined tag, enter a unique tag name. To add a value to an existing predefined tag, specify the name of the existing tag and

the new value. This cmdlet returns an object that represents the new or modified tag with its values and the number of resources to which it has been applied. The

Azure Tags module that New-AzTag is part of can help you manage predefined Azure tags. An Azure tag is a name-value pair that you can use to categorize your Azure

resources and resource groups, such as by department or cost center, or to track notes or comments about the resources and groups. You can define and apply tags in a

single step, but predefined tags let you establish standard, consistent, predictable names and values for the tags in your subscription. To apply a predefined tag to

a resource or resource group, use the Tag parameter of the New-AzTag cmdlet. To search for resource groups with a specified tag name or name and value, use the Tag

parameter of the Get-AzResourceGroup cmdlet. Every tag has a name. The values are optional. A predefined Azure tag can have multiple values, but when you apply the

tag to a resource or resource group, you apply the tag name and only one of its values. For example, you can create a predefined Department tag with a value for each

department, such as Finance, Human Resources, and IT. When you apply the Department tag to a resource, you apply only one predefined value, such as Finance.

CreateByResourceIdParameterSet : The New-AzTag cmdlet with a ResourceId creates or updates the entire set of tags on a resource or subscription. This operation allows

adding or replacing the entire set of tags on the specified resource or subscription. The specified entity can have a maximum of 50 tags.

PARAMETERS

-DefaultProfile <Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>

The credentials, account, tenant, and subscription used for communication with azure.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Name <System.String>

Specifies the predefined tag name. To create a new predefined tag, enter a unique name. To add a value to an existing tag, enter the name of the existing tag. If

an existing predefined tag has the specified name, New-AzTag adds the specified value, if any, to the tag with that name instead of creating a new tag.

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-ResourceId <System.String>

The resource identifier for the entity being tagged. A resource, a resource group or a subscription may be tagged.

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Tag <System.Collections.Hashtable>

The tags to put on the resource.

Required? true

Position? 1

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Value <System.String>

Specifies a predefined tag value. Predefined tags can have multiple values, but you can enter only one value in each command. This parameter is optional, because tags can have names without values.

Required?	false
Position?	1
Default value	None
Accept pipeline input?	True (ByPropertyName)
Accept wildcard characters?	false

-Confirm <System.Management.Automation.SwitchParameter>

Prompts you for confirmation before running the cmdlet.

Required?	false
Position?	named
Default value	False
Accept pipeline input?	False
Accept wildcard characters?	false

-WhatIf <System.Management.Automation.SwitchParameter>

Shows what would happen if the cmdlet runs. The cmdlet is not run.

Required?	false
Position?	named
Default value	False
Accept pipeline input?	False
Accept wildcard characters?	false

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<https://go.microsoft.com/fwlink/?LinkID=113216>).

INPUTS

System.String

System.Collections.Hashtable

OUTPUTS

Microsoft.Azure.Commands.ResourceManager.Common.Tags.PSTag |

Microsoft.Azure.Commands.Tags.Model.PSTagResource

NOTES

----- Example 1: Create a predefined tag -----

New-AzTag -Name "FY2015"

Name	ValuesTable	Count	Values
----	-----	-----	-----
FY2015	0	{}	

This command creates a predefined tag named FY2015. This tag has no values. You can apply a tag with no values to a resource or resource group, or use New-AzTag to add values to the tag. You can also specify a value when you apply the tag to the resource or resource group.

----- Example 2: Create a predefined tag with a value -----

New-AzTag -Name "Department" -Value "Finance"

Name: Department

Count: 0

Values:

Name	Count
Finance	0

This command creates a predefined tag named Department with a value of Finance.

----- Example 3: Add a value to a predefined tag -----

New-AzTag -Name "Department" -Value "Finance"

Name: Department

Count: 0

Values:

Name	Count
Finance	0

New-AzTag -Name "Department" -Value "IT"

Name: Department

Count: 0

Values:

Name	Count
Finance	0
IT	0

These commands create a predefined tag named Department with two values. If the tag name exists, New-AzTag adds the value to the existing tag instead of creating a new one.

----- Example 4: Use a predefined tag -----

```
New-AzTag -Name "CostCenter" -Value "0001"
```

Name: CostCenter

Count: 0

Values:

Name	Count
=====	=====
0001	0

```
Set-AzResourceGroup -Name "EngineerBlog" -Tag @{Name="CostCenter";Value="0001"}
```

Name: EngineerBlog

Location: East US

Resources:

Name	Type	Location
=====	=====	=====
EngineerBlog	Microsoft.Web/sites	West US
EngSvr01	Microsoft.Sql/servers	West US
EngDB02	Microsoft.Sql/databases	West US

Tags:

Name	Value
=====	=====
CostCenter	0001

Get-AzTag -Name "CostCenter"

Name: CostCenter

Count: 1

Values:

Name	Count
0001	1

Get-AzResourceGroup -Tag @{Name="CostCenter"}

Name: EngineerBlog

Location: East US

Resources:

Name	Type	Location
EngineerBlog	Microsoft.Web/sites	West US
EngSvr01	Microsoft.Sql/servers	West US
EngDB02	Microsoft.Sql/databases	West US

Tags:

Name	Value
CostCenter	0001

The commands in this example create and use a predefined tag.

Example 5: Creates or updates the entire set of tags on a subscription

\$Tags = @{"tagKey1"="tagValue1"; "tagKey2"="tagValue2"}

New-AzTag -ResourceId /subscriptions/xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx -Tag \$Tags

Id : {Id}
Name : {Name}
Type : {Type}
Properties :

Name	Value
=====	=====
tagKey1	tagValue1
tagKey2	tagValue2

This command creates or updates the entire set of tags on the subscription with {subId}.

Example 6: Creates or updates the entire set of tags on a resource

\$Tags = @{"Dept"="Finance"; "Status"="Normal"}

```
New-AzTag -ResourceId /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/resourcegroups/testrg/providers/Microsoft.Sql/servers/Server1 -Tag $Tags
```

Id : {Id}
Name : {Name}
Type : {Type}
Properties :

Name	Value
=====	=====
Dept	Finance
Status	Normal

This command creates or updates the entire set of tags on the resource with {resourceId}.

Online Version: <https://learn.microsoft.com/powershell/module/az.resources/new-aztag>

Get-AzTag

Remove-AzTag

Update-AzTag